The source file name consists of the case-sensitive name of the top-level class it contains (of which there is exactly one), plus the .java extension.

Source files are encoded in UTF-8.

Camel cases is used

de from the line terminator sequence, the ASCII horizontal space character (0x20) is the only whitespace character that appears anywhere in a source file. This implies that:

1. All other whitespace characters in string and character literals are escaped.
2. Tab characters are not used for indentation.

A source file consists of, in order:

1. License or copyright information, if present
2. Package statement
3. Import statements
4. Exactly one top-level class
   Exactly one blank line separates each section that is present.

The package statement is not line-wrapped

Wildcard imports, static or otherwise, are not used.

Imports are ordered as follows:

1. All static imports in a single block.
2. All non-static imports in a single block.
   If there are both static and non-static imports, a single blank line separates the two blocks. There are no other blank lines between import statements.

Within each block the imported names appear in ASCII sort order.
Static import is not used for static nested classes. They are imported with normal imports.

The order you choose for the members and initializers of your class can have a great effect on learnability. However, there's no single correct recipe for how to do it; different classes may order their contents in different ways.

What is important is that each class uses *some* logical order, which its maintainer could explain if asked. For example, new methods are not just habitually added to the end of the class, as that would yield "chronological by date added" ordering, which is not a logical ordering.

3.4.2.1 Overloads: never split

When a class has multiple constructors, or multiple methods with the same name, these appear sequentially, with no other code in between (not even private members).